

Listing of the Claims

1. (Previously Presented) A computer-implemented method comprising:
generating a customizable product configurator, said generating comprising
creating a customizable product class, wherein
the customizable product class comprises a set of one or more
attributes to define the customizable product class;
adding a component product class to the customizable product class,
wherein
the component product class is a subclass of the customizable
product class, and
the component product class comprises customizable class rules;
and
mapping a customizable user interface (UI) to the customizable product
class, wherein
the customizable UI provides an access structure to the
customizable product configurator,
elements of the customizable UI access structure correspond to
elements of the customizable product class,
the customizable product class is configured to represent a
customizable product, and
the component product class is configured to represent one or more
components of the customizable product.
2. (Original) The method of claim 1 wherein the component product class
includes component product subclasses.
3. (Original) The method of claim 1 wherein the component product class
inherits the attributes of the customizable product class.

4. (Previously Presented) The method of claim 1 wherein said generating further comprises:
adding one or more component product classes to a port; and
adding the port to the customizable product class, wherein
the port allows the configurator to classify a group of component products.
5. (Previously Presented) The method of claim 4, wherein
the port comprises a cardinality attribute, and
the cardinality attribute constrains the number of component products to be added
by the configurator.
6. (Previously Presented) The method of claim 5, wherein
the cardinality attribute comprises a minimum cardinality and a maximum
cardinality,
the minimum cardinality constrains the minimum number of component products
to be added by the configurator, and
the maximum cardinality constrains the maximum number of component products
to be added by the configurator.
7. (Previously Presented) The method of claim 5, wherein
the cardinality attribute comprises a default cardinality, and
the default cardinality defines a quantity of the component product class added by
the configurator.
8. (Previously Presented) The method of claim 1 wherein the mapping
comprises:
building the customizable UI from a set of themes, groups, and controls.
9. (Previously Presented) The method of claim 8 wherein a theme of the set
of themes, groups, and controls comprises one or more of tabs and wizards.

10. (Previously Presented) The method of claim 8 wherein each theme in the set of themes, groups, and controls comprises at least one of the set of background colors, fonts, and multi-linguals.

11. (Canceled)

12. (Previously Presented) The method of claim 8 wherein a control of the set of themes, groups, and controls comprises one or more of a drop down box, a radio button, and a list box.

13. (Previously Presented) The method of claim 1 further comprising:
generating a user interface for the component product class using the
customizable UI.

14. (Previously Presented) The method of claim 1 wherein the customizable UI is a subclass of the customizable product class.

15. (Previously Presented) The method of claim 1 further comprising:
generating a configurator user interface with HTML, Applets, and ActiveX
programming languages, wherein
said generating uses the customizable UI.

16. (Previously Presented) The method of claim 1, wherein
the component product class comprises a static attribute, and
the static attribute is not associated with a parent class.

17. (Previously Presented) The method of claim 1 wherein the component product class, customizable class rules, and customizable UI are object oriented classes.

18. (Original) The method of claim 1 wherein the customizable product has an object oriented structure.

19. (Original) The method of claim 1 wherein the customizable product includes versioning.

20. (Original) The method of claim 1 wherein the configurator is stored in a data store.

21. (Previously Presented) A computer-readable storage medium that provides instructions, which when executed by a set of one or more processors, cause the set of processors to perform operations comprising:

generating a customizable product configurator, said generating comprising

creating a customizable product class, wherein

the customizable product class comprises a set of one or more

attributes to define the customizable product class;

adding a component product class to the customizable product class,

wherein

the component product class is a subclass of the customizable product class, and

the component product class comprises customizable class rules;

and

mapping a customizable UI to the customizable product class, wherein

the customizable UI provides an access structure to the configurator,

elements of the customizable UI access structure correspond to

elements of the customizable product class,

the customizable product class is configured to represent a customizable product, and

the component product class is configured to represent one or more components of the customizable product.

22. (Previously Presented) The computer-readable storage medium of claim 21 wherein the component product class includes component product subclasses.

23. (Previously Presented) The computer-readable storage medium of claim 21 wherein the component product class inherits the attributes of the customizable product class.

24. (Previously Presented) The computer-readable storage medium of claim 21 further provides instructions that cause the set of processors to perform operations comprising:

- adding one or more component product classes to a port; and
- adding the port to the customizable product class, wherein
the port allows the configurator to classify a group of component products.

25. (Previously Presented) The computer-readable storage medium of claim 24, wherein

- the port comprises a cardinality attribute, and
- the cardinality attribute constrain the number of component products to be added
by the configurator.

26. b(Previously Presented) The computer-readable storage medium of claim 25, wherein

- the cardinality attribute comprises a minimum cardinality and a maximum
cardinality,
- the minimum cardinality constrains the minimum number of component products
to be added by the configurator, and
- the maximum cardinality constrains the maximum number of component products
to be added by the configurator.

27. (Previously Presented) The computer-readable storage medium of claim 25, wherein

- the cardinality attribute comprises a default cardinality, and
- the default cardinality defines a quantity of the component product class added by
the configurator.

28. (Previously Presented) The computer-readable storage medium of claim 21 further providing instructions for the mapping comprising:

building the customizable UI from a set of themes, groups, and controls.

29. (Previously Presented) The computer-readable storage medium of claim 28 wherein a theme of the set of themes, groups, and controls comprises one or more of tabs and wizards.

30. (Previously Presented) The computer-readable storage medium of claim 28 wherein a theme of the set of themes, groups, and controls comprises one or more of background colors, fonts, and multi-linguals.

31. (Canceled)

32. (Previously Presented) The computer-readable storage medium of claim 28 wherein a control of the set of themes, groups, and controls comprises one or more of a drop down box, a radio button, and a list box.

33. (Previously Presented) The computer-readable storage medium of claim 21 that further provides instructions causing the set of processors to perform operations comprising:

generating a user interface for the component product class using the customizable UI.

34. (Previously Presented) The computer-readable storage medium of claim 21 wherein the customizable UI is a subclass of the customizable product class.

35. (Previously Presented) The computer-readable storage medium of claim 21 that further provides instructions causing the set of processors to perform operations comprising:

generating a configurator user interface with HTML, Applets, and ActiveX programming languages, wherein said generating uses the customizable UI.

36. (Previously Presented) The computer-readable storage medium of claim 21, wherein

the component product class comprises a static attribute, and
the static attribute is not associated with a parent class.

37. (Previously Presented) The computer-readable storage medium of claim 21 wherein the component product class, customizable class rules, and customizable UI **class** are object oriented classes.

38. (Previously Presented) The computer-readable storage medium of claim 21 wherein the customizable product has an object oriented structure.

39. (Previously Presented) The computer-readable storage medium of claim 21 wherein the customizable product includes versioning.

40. (Previously Presented) The computer-readable storage medium of claim 21 wherein the configurator is stored in a data store.

41. (Previously Presented) An object oriented configurator comprising:
a processor;

a memory coupled to the processor;

a customizable product class stored in the memory;

a component product class, wherein

the component product class is a subclass of the customizable product
class, and

the component product class inherits a set of one or more attributes from
the customizable product class; and

a customizable UI, wherein

the customizable UI is mapped to the customizable product class

providing a view of the component product,

the customizable UI provides an access structure,

the customizable product class is configured to represent a customizable product, and
the component product class is configured to represent one or more components of the customizable product.

42. (Previously Presented) The object oriented configurator in claim 41 further comprising:

a port comprising a set of one or more component products of the component product class.

43. (Previously Presented) The object oriented configurator in claim 42, wherein

the port comprises a cardinality, and
the cardinality constrains the number of component products to add to the customizable product class.

44. (Previously Presented) The object oriented configurator in claim 43, wherein

the cardinality attribute comprises a minimum cardinality and a maximum cardinality,
the minimum cardinality constrains the minimum number of component products to be added by the configurator, and
the maximum cardinality constrains the maximum number of component products to be added by the configurator.

45. (Previously Presented) The object oriented configurator in claim 43, wherein

the cardinality comprises a default cardinality, and
the default cardinality defines a quantity of the component product class added by the configurator.

46. (Previously Presented) The object oriented configurator in claim 41 wherein the customizable class rule, and customizable UI are subclasses of the customizable product class.

47. (Previously Presented) The object oriented configurator in claim 41, wherein
the component product class comprises a static attribute, and
the static attribute is not inherited from a parent class.

48. (Original) The object oriented configurator in claim 41 wherein the attribute is of type string, number, date, and Boolean.

49. (Previously Presented) The object oriented configurator in claim 41 further comprising a second customizable product class.

50. (Previously Presented) The object oriented configurator in claim 49 wherein the second customizable product class comprises one or more component products of the component product class.

51. (Previously Presented) The object oriented configurator in claim 41 wherein a component product of the component product class includes an expression to restrict the component product from becoming a subclass of the customizable product class.

52. (Previously Presented) The object oriented configurator in claim 41 further comprising:
a script configured to communicate with another application.

53. (Original) The object oriented configurator in claim 41 wherein the customizable UI includes a theme, group, and control.

54. (Previously Presented) The object oriented configurator in claim 41 wherein the theme includes one or more of a tab, wizard, font, and color.

55. (Original) The object oriented configurator in claim 41 wherein the control includes one or more of a drop down box, a radio button, and a list box.

56. (Original) The object oriented configurator in claim 41 wherein the customizable UI map comprises HTML, JAVA applets, and ActiveX components.

57. (Original) The object oriented configurator in claim 41 wherein each component product class has an unique identifier, the unique identifier is used to locate an associative component product.

58. (Original) The object oriented configurator in claim 41 further comprising link items.

59. (Previously Presented) An apparatus comprising:

a processor;

means for generating a customizable product configurator using the processor,

said means for generating comprising

means for creating a customizable product class, wherein

the customizable product class comprises a set of one or more

attributes to define the customizable product class;

means for adding a component product class to the customizable product

class, wherein

the component product class is a subclass of the customizable

product class, and

the component product class comprises customizable class rules;

and

means for mapping a customizable UI to the customizable product class,

wherein

the customizable UI provides an access structure to the

configurator,

elements of the customizable UI access structure correspond to

elements of the customizable product class,

the customizable product class is configured to represent a customizable product, and
the component product class is configured to represent one or more components of the customizable product.

60. (Previously Presented) The apparatus of claim 59 wherein the component product class includes component product subclasses.

61. (Previously Presented) The apparatus of claim 59 wherein the component product class inherits the attributes of the customizable product class.

62. (Previously Presented) The apparatus of claim 59 wherein said means for generating further comprises:
means for adding one or more component product classes to a port; and
means for adding the port to the customizable product class, wherein
the port allows the configurator to classify a group of component products.

63. (Previously Presented) The apparatus of claim 62, wherein
the port comprises a cardinality attribute, and
the cardinality attribute constrains the number of component products to be added
by the configurator.

64. (Previously Presented) The apparatus of claim 63, wherein
the cardinality attribute comprises a minimum cardinality and a maximum
cardinality,
the minimum cardinality constrains the minimum number of component products
to be added by the configurator, and
the maximum cardinality constrains the maximum number of component products
to be added by the configurator.

65. (Previously Presented) The apparatus of claim 63, wherein
the cardinality attribute comprises a default cardinality, and

the default cardinality defines a quantity of the component product class added by the configurator.

66. (Previously Presented) The apparatus of claim 59 wherein the means for mapping comprises:

means for building the customizable UI from a set of themes, groups, and controls.

67. (Previously Presented) The apparatus of claim 66 wherein a theme of the set of themes, groups, and controls comprises one or more of tabs and wizards.

68. (Previously Presented) The apparatus of claim 66 wherein each theme in the set of themes, groups, and controls comprises at least one of a set of background colors, fonts, and multi-linguals.

70. (Previously Presented) The apparatus of claim 66 wherein a control of the set of themes, groups, and controls comprises one or more of a drop down box, a radio button, and a list box.

71. (Previously Presented) The apparatus of claim 59 wherein the means for generating further comprises:

means for generating a user interface for a component product class using the customizable UI.

72. (Previously Presented) The apparatus of claim 59 wherein the customizable UI is a subclass of the customizable product class.

73. (Previously Presented) The apparatus of claim 59 wherein the means for generating further comprises:

means for generating a configurator user interface with HTML, Applets, and ActiveX programming languages, wherein the means for generating uses the customizable UI.

74. (Previously Presented) The apparatus of claim 59, wherein the component product class comprises a static attribute, and the static attribute is not associated with a parent class.

75. (Previously Presented) The apparatus of claim 59 wherein the component product class, customizable class rules, and customizable UI ~~class~~ are object oriented classes.

76. (Previously Presented) The apparatus of claim 59 wherein the customizable product has an object oriented structure.

77. (Previously Presented) The apparatus of claim 59 wherein the customizable product includes versioning.

78. (Previously Presented) The apparatus of claim 59 wherein the configurator is stored in a data store.